

# Abuse / Anomaly Detection in Text Messages Using Python

Prachi Ghoghari

Email: [prachipatel0848@gmail.com](mailto:prachipatel0848@gmail.com)

GitHub: <https://github.com/prachi01645>

Date: January 2026

## 1. Abstract

This project implements a simple, self-contained Python program to detect abusive or anomalous text messages. It flags messages containing predefined abusive words and summarizes the percentage of abusive content. The project demonstrates practical skills in text processing, logical reasoning, and independent experimentation.

## 2. Introduction

The proliferation of online communication has increased the need for automated systems to detect abusive or harmful content. This project simulates a lightweight abuse detection system that can serve educational purposes or act as a prototype for larger applications.

## 3. Methodology

1. **Dataset:** A small set of simulated text messages representing typical safe and abusive content.
2. **Abusive Word List:** Predefined list of words indicative of abusive language.
3. **Detection Logic:** Each message is scanned; if it contains any word from the abusive list, it is flagged as "Abusive," otherwise "Safe."
4. **Summary Metrics:** The program outputs the total number and percentage of messages flagged as abusive.
5. **Implementation:** Entirely in pure Python, requiring no external libraries.

## 4. Result

Abuse Detection Output:

Safe: You are amazing!

Abusive: I hate you!

Abusive: You are stupid

Safe: Have a great day!

Abusive: This is terrible =>Summary: 3/5 messages flagged as abusive (60.0%)

## Abuse Detection Output:

Safe: You are amazing!

Abusive: I hate you!

Abusive: You are stupid

Safe: Have a great day!

Abusive: This is terrible

Summary: 3/5 messages flagged as abusive (60.0%)

- “Safe” = no abusive words detected
- “Abusive” = message contains words from the predefined list

## 5.Observations

- The detection is accurate for the provided word list.
- Demonstrates the relationship between keyword-based detection and overall content safety.
- Highlights the importance of automated systems in monitoring textual communication.

## 6.Applications

- **Educational:** Learn Python scripting and text processing techniques.
- **Research:** Prototype for experimenting with more sophisticated abuse detection algorithms.
- **Industry:** Can be extended for moderation systems in social media, chat applications, or online forums.

## 7.Future Work

- Expand to **larger and real-world datasets**.
- Implement **more sophisticated detection techniques**: machine learning models, NLP, sentiment analysis.
- Add **visual dashboards** to monitor abusive content in real time.
- Support **multi-language detection** for global applications.

## 8.Conclusion

This project demonstrates an independent implementation of an abuse/anomaly detection system in Python. It is fully self-contained, easy to run, and provides a foundation for learning, experimentation, and further development in automated content monitoring.

## 9.Skills Demonstrated

- Python programming (pure, no dependencies)
- Logical reasoning and text processing
- Independent experimentation and reproducibility

## 10.Further Observation

- This project demonstrates **independent problem-solving** by creating a functional prototype without external libraries.
- The rule-based detection approach highlights **trade-offs between simplicity and scalability** in real-world systems.
- Provides a foundation for exploring **more advanced text analytics** in the future.
- Shows **practical application of Python for ethical AI projects** and content moderation.
- Can be used as a **teaching tool** to illustrate how ML/NLP concepts can be applied in simple yet effective ways.
- Encourages **further experimentation**, like expanding the dataset, adding patterns, or integrating with ML models.